Smartcards e Cartão de Cidadão

Smartcards

- **Dispositivos físicos para armazenamento de chaves e operações sobre as mesmas**
  - Invioláveis, resistentes a ataques por canais paralelos ou vírus

- **Objetivo: permitir a utilização de chaves, sem o seu compromisso**
  - Titular pode utilizar chave para realizar operações criptográficas (Simétricas e assimétricas)
    - Autenticar o titular, Gerar assinaturas de documentos, Gerar respostas a desafios, Armazenar valores
- **Utilizações:**
  - Autenticação, Cartões bancários, Cartões de Identificação, Transportes, SIM

Smartcards

- **Cartão com capacidades de computação**
  - CPU
  - ROM
  - EEPROM
  - RAM
- **Interface**
  - Com contactos
  - Sem contactos

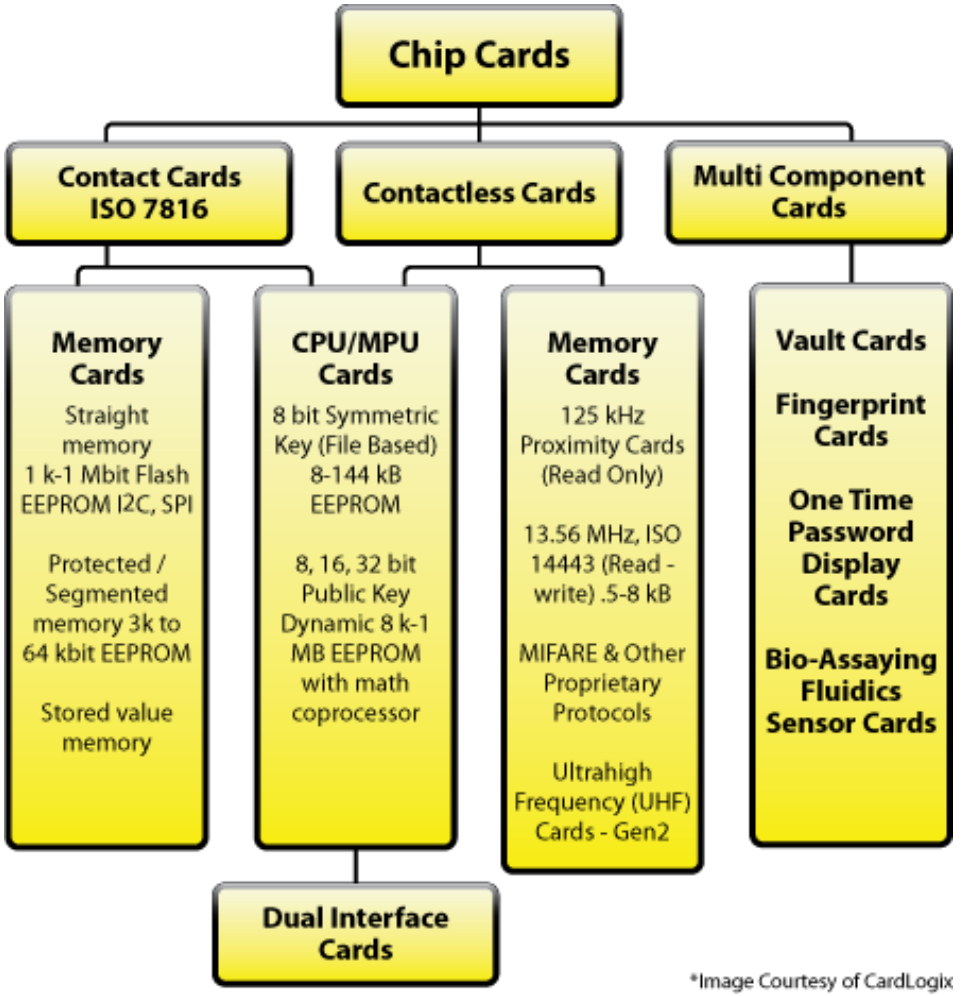
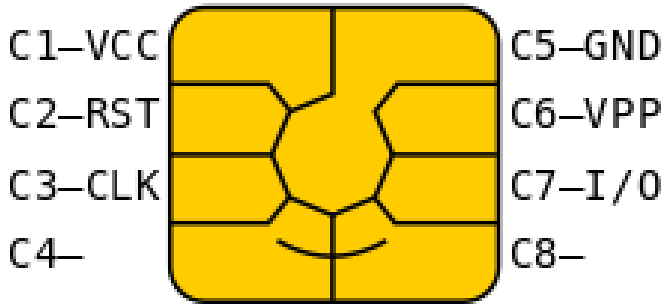


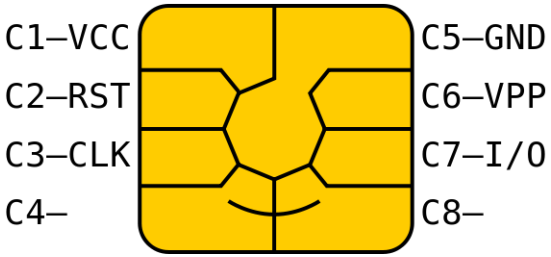
*Image Courtesy of CardLogix

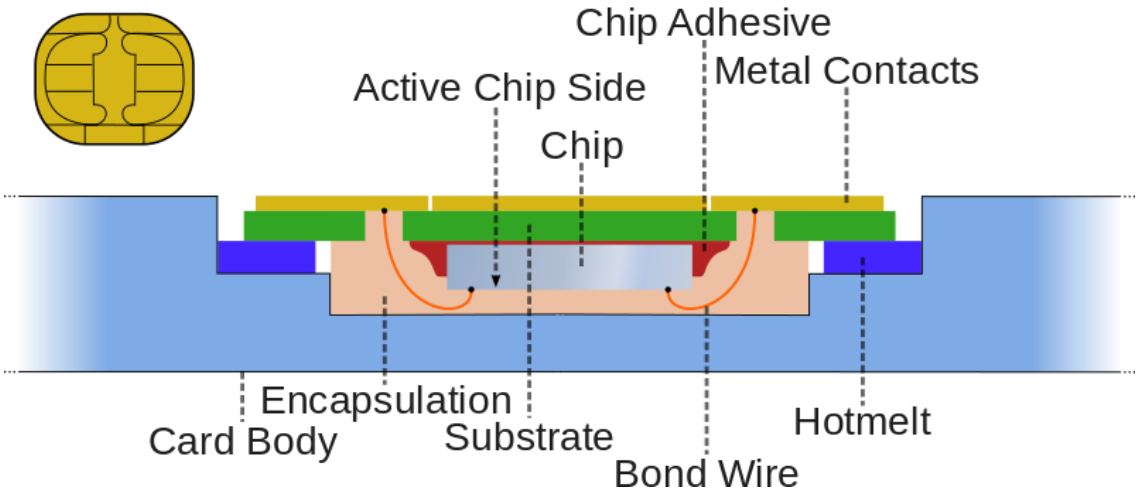
Smartcards

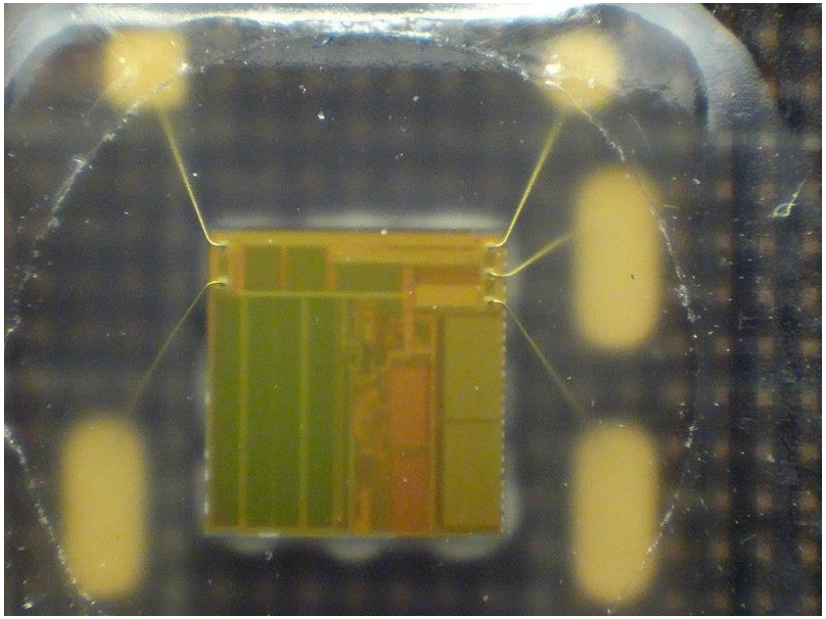
- **CPU**
  - 8/16 bit
  - Crypto-coprocessor (opt.)
- **ROM**
  - Sistema Operativo
  - Comunicação
  - Algoritmos criptográficos
- **EEPROM**
  - Sistema de Ficheiros
    - Programas / aplicações
    - Chaves/ passwords
- **RAM**
  - Dados temporários
    - Apagados quando cartão é desligado
- **Contactos Mecânicos**
  - ISO 7816-2



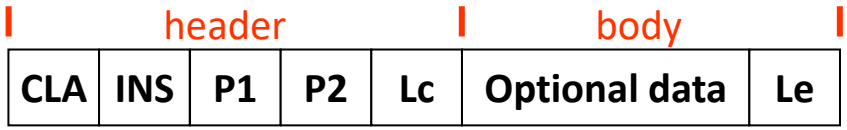
- **Segurança Física**
  - Resistente a acessos físicos diretos
  - Resistente a ataques por canais paralelos


C1–VCC
C2–RST
C3–CLK
C4–
C5–GND
C6–VPP
C7–I/O
C8–


Chip Adhesive
Metal Contacts
Active Chip Side
Chip
Encapsulation
Card Body
Substrate
Bond Wire
Hotmelt



Interação com Smartcards: APDU (ISO 7816-4)

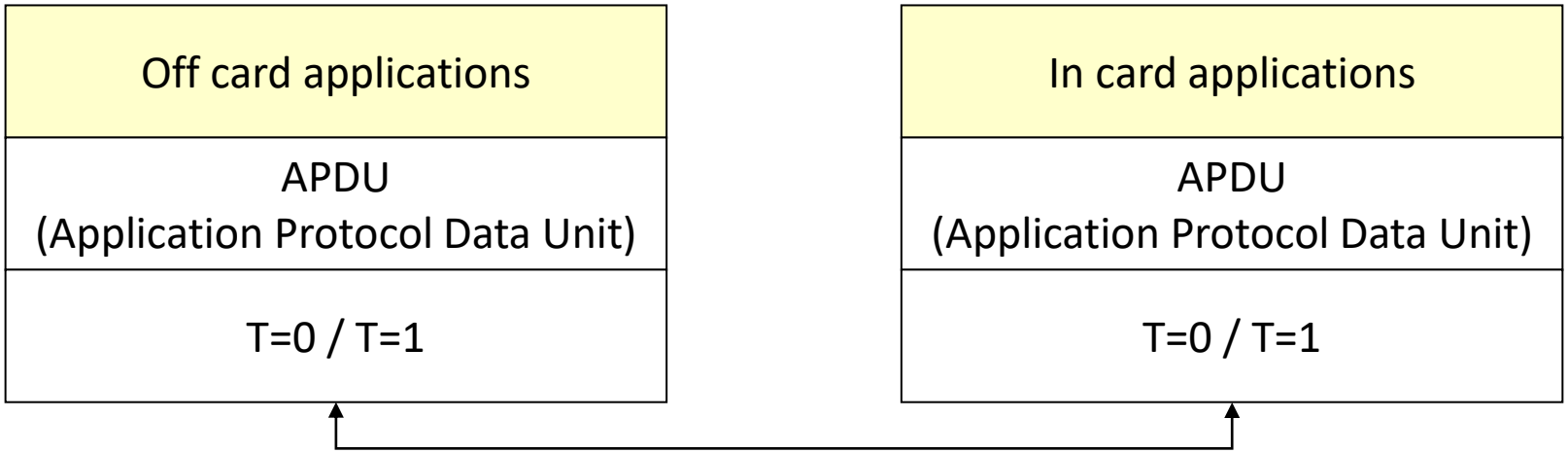


- **APDU de Comando**
  - CLA (1 byte)
    - Classe da instrução
  - INS (1 byte)
    - Comando
  - P1 e P2 (2 bytes)
    - Parâmetros específicos do comando
  - Lc
    - Comprimento dos dados opcionais
  - Le
    - Comprimento dos dados esperados na resposta
    - Zero (0) significa todos os dados disponíveis
- **APDU de Resposta**
  - SW1 e SW2 (2 bytes)
    - Byte de estado
    - 0x9000 significa SUCESSO

Interação com o Smartcard: Protocolos de baixo-nível T=0 e T=1

- **T=0**
  - Enviado um octeto de cada vez
  - Mais lento
- **T=1**
  - Octetos transmitidos em blocos
  - Mais rápido mas requer suporte nas camadas superiores
- **ATR (ISO 7816-3)**
  - Resposta à operação de RESET
  - Reporta o protocolo esperado pelo cartão

Pilha de Comunicações



Interação com o Smartcard: Protocolos de baixo-nível T=0 e T=1

```
ATR: 3B 7D 95 00 00 80 31 80 65 B0 83 11 00 C8 83 00 90 00
+ TS = 3B --> Direct Convention
+ T0 = 7D, Y(1): 0111, K: 13 (historical bytes)
  TA(1) = 95 --> Fi=512, Di=16, 32 cycles/ETU
    125000 bits/s at 4 MHz, fMax for Fi = 5 MHz => 156250 bits/s
  TB(1) = 00 --> VPP is not electrically connected
  TC(1) = 00 --> Extra guard time: 0
+ Historical bytes: 80 31 80 65 B0 83 11 00 C8 83 00 90 00
  Category indicator byte: 80 (compact TLV data object)
    Tag: 3, len: 1 (card service data byte)
      Card service data byte: 80
        - Application selection: by full DF name
        - EF.DIR and EF.ATR access services: by GET RECORD(s) command
        - Card with MF
    Tag: 6, len: 5 (pre-issuing data)
      Data: B0 83 11 00 C8
    Tag: 8, len: 3 (status indicator)
      LCS (life card cycle): 00 (No information given)
      SW: 9000 (Normal processing.)

Possibly identified card (using /usr/share/pcsc/smartcard_list.txt):
3B 7D 95 00 00 80 31 80 65 B0 83 11 00 C8 83 00 90 00
3B 7D 95 00 00 80 31 80 65 B0 83 11 .. .. 83 00 90 00
        Portuguese ID Card (eID)
        http://www.cartaodecidadao.pt/
```

Codificação de objetos nos smartcards: TLV e ASN.1 BER

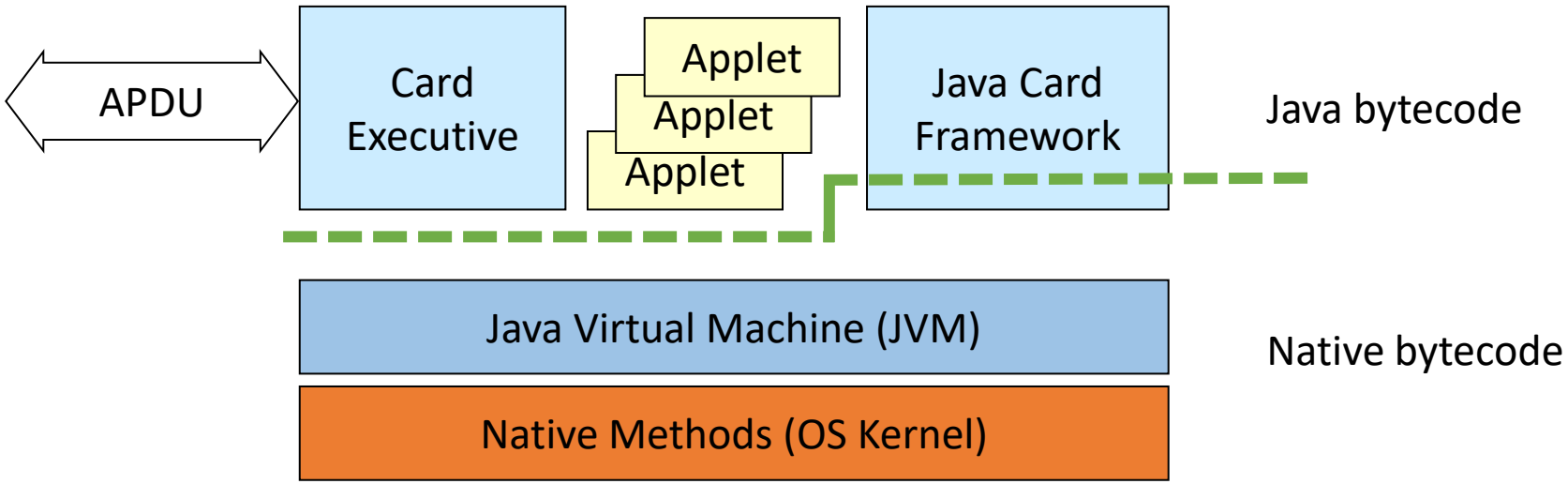
- **Tag-Length-Value (TLV)**
  - Tag: Tipo de objeto
  - Length: Tamanho do objeto
  - Value: Dados do objeto
- **Cada TLV é codificado através das regras ASN.1 BER**
  - Abstract Syntax Notation, Basic Encoding Rules
- **Dados de um objeto podem conter outros TLV**
  - Estrutura recursiva
- **Permite ignorar objetos desconhecidos**

Modelo de computação do Smartcard
Cartões Java

- **Smartcards executam Applets Java**
  - Utilizam o Java Card Runtime Environment

- **O JCRE executa no topo do SO nativo**
  - Java Virtual Machine
  - Card Executive
    - Gestão do Cartão
    - Comunicações
  - Java Card Framework
    - Bibliotecas de funções

Modelo de computação do Smartcard
Cartões Java



Cartão de Cidadão

- **Cartão de identificação das dimensões de um cartão de crédito**
- **Contém vários métodos de fornecer informação identidade**
  - Informática
    - Interação com o Smartcard
  - Visual, legível por humanos
    - Fotografia, números e nomes
  - Visual, legível por dispositivos
    - MRZ (Machine Readable Zone)

Atributos Visuais: Legíveis por humanos

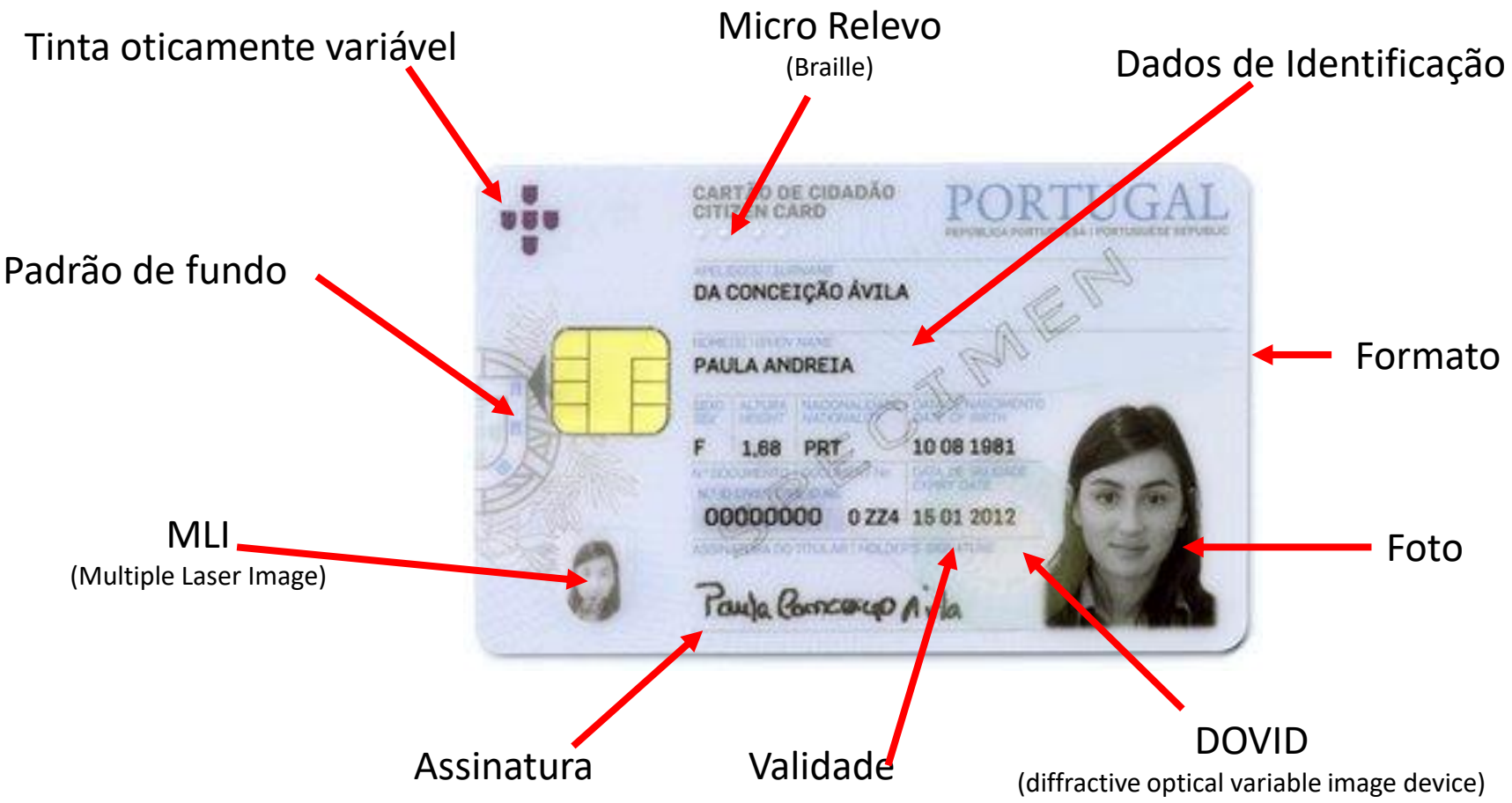
- **Nome**
  - Sobrenome, Nome próprio, Pais
- **Atributos físicos**
  - Sexo e Altura
- **Outros**
  - Data de nascimento, nacionalidade
  - Fotografia
  - Assinatura caligráfica
- **Números**
  - Número de identificação Civil (e checksum)
  - Num: Identificação Fiscal, Sistema Nacional de Saúde, Segurança Social
  - Número do documento e validade
- **Versão do cartão**

Atributos visuais: legíveis por dispositivos

- **Nome**
  - Sobrenome, Nome próprio, Nomes adicionais
  - Número de nomes
- **Atributos Físicos**
  - Sexo
- **Outros**
  - Data de nascimento e nacionalidade
- **Números**
  - Identificação Civil (e checksum)
  - Número do documento (e checksum)
  - Número de documentos emi
- **Validade**


```
I<PRT000000000<<0<ZZ4<<<<<<<<2
8108108F1201158PRT<<<<<<<<<<<1
AVILA<<PAULA<ANDREIA<CONCEICAO
```

Atributos Visuais de Segurança



Atributos Digitais

- **Todos os atributos visíveis com a exceção da assinatura**
- **Morada**
- **Modelo da impressão digital biométrica**
- **2 pares de chaves assimétricos (Autenticação e Assinatura)**
- **5 certificados de chave pública**
  - 2 relacionados com os pares de chaves anteriores
  - 3 relacionadas a CAs intermédias necessárias para construir o caminho de certificação
- **1 chave simétrica para EMV-CAP (retirado recentemente)**
- **4 Códigos de utilizadores (PINs)**
  - Autenticação, Assinatura, Morada, PUK

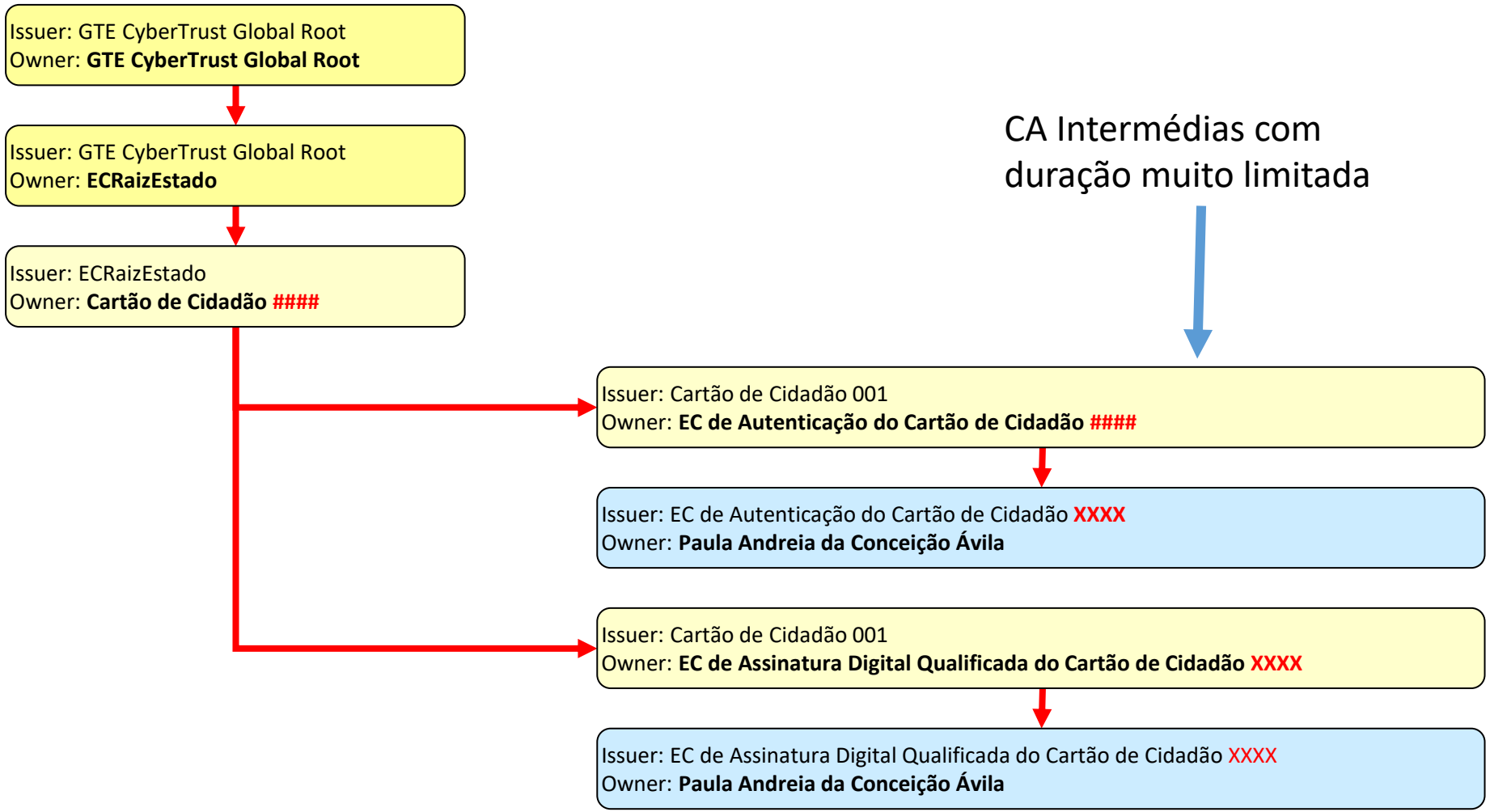
Proteção por PIN

- **Possuir o cartão é insuficiente para**
  - Obter morada (exceto nos recentes)
  - Obter ou usar a chave privada de autenticação
  - Obter ou usar a chave privada de assinatura
  - Obter ou usar a chave secreta de EMV-CAP
- **Operações protegidas por PIN**
  - PIN de 4 números
  - PIN é bloqueado após 3 tentativas incorretas
- **Exceções**
  - Forças policiais podem obter a morada sem o PIN

Certificados no Smartcard: Objetivos

- **Possibilita autenticar o dono do cartão**
  - O dono pode distribuir o seu certificado para outras pessoas/serviços que passar a poder verificar a sua identidade
- **Possibilita o dono autenticar outras pessoas com cartões semelhantes**
  - Cadeia de certificação presente no cartão
- **Possibilita o cartão autenticar clientes com certificados semelhantes**
  - Algumas operações podem ser pedidas ao cartão com certificados "especiais" que o cartão valida

Certificados no Smartcard



Certificados no Smartcard: Interoperação com outras aplicações

Aplicações de watchdog detetam inserção e remoção

- **Inserção**
  - Aplicações obtêm certificados e inserem-nos nos repositórios dos navegadores
  - Utilização das chaves respetivas é condicionada pelos PIN
- **Remoção**
  - Aplicações removem certificados dos repositórios dos navegadores

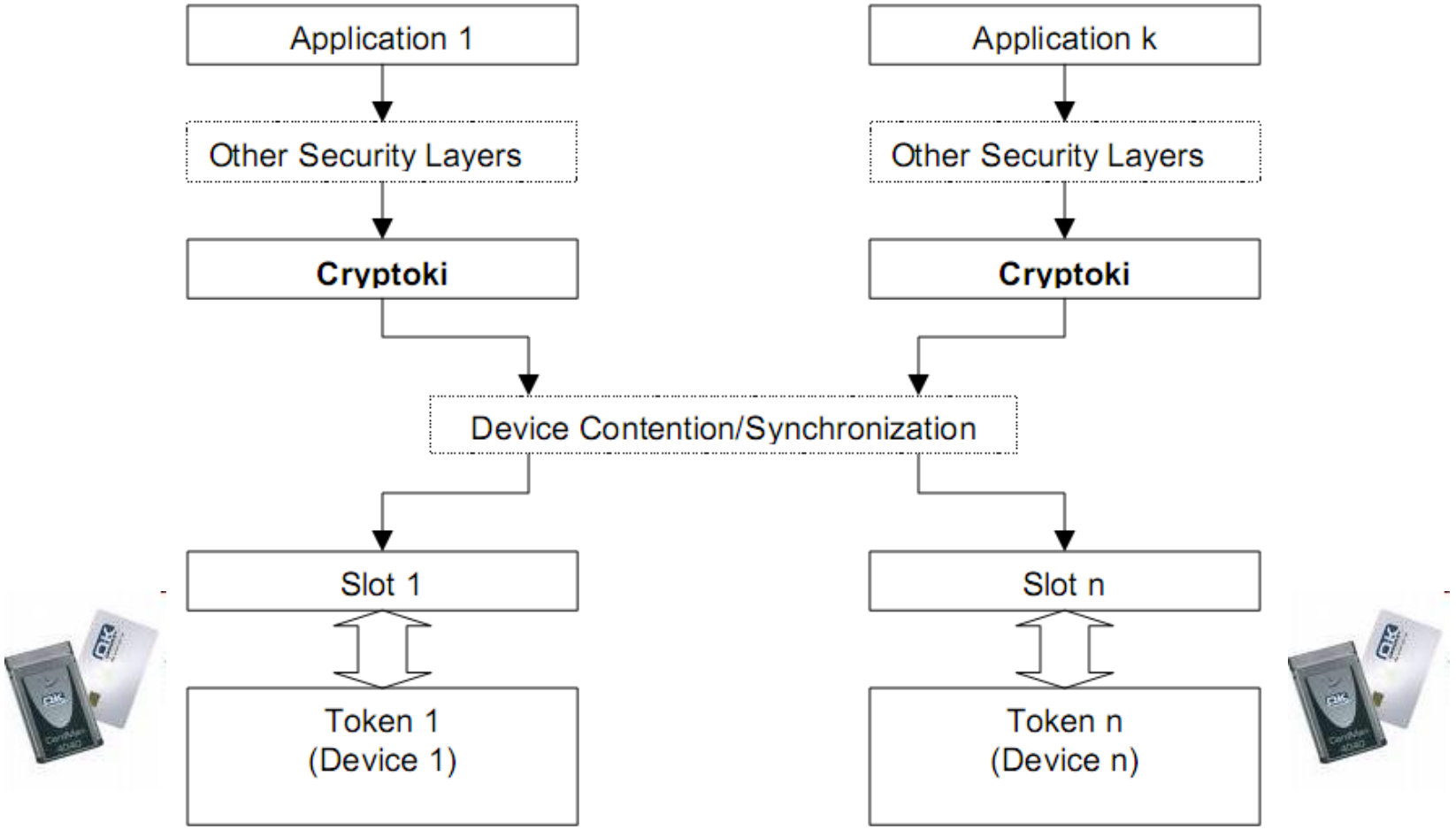
Aplicações em Smartcards: Aplicações no Cartão de Cidadão

- **IAS Classic V3**
  - Autenticação e assinatura digital
  - Utilização de pares de chaves assimétricas
- **EMV-CAP**
  - Geração de one-time-passwords para canais alternativos (telefone, Fax, etc..)
  - Retirado em 2016
- **Precise Biometric BIO Match On Card**
  - Validação de impressões digitais

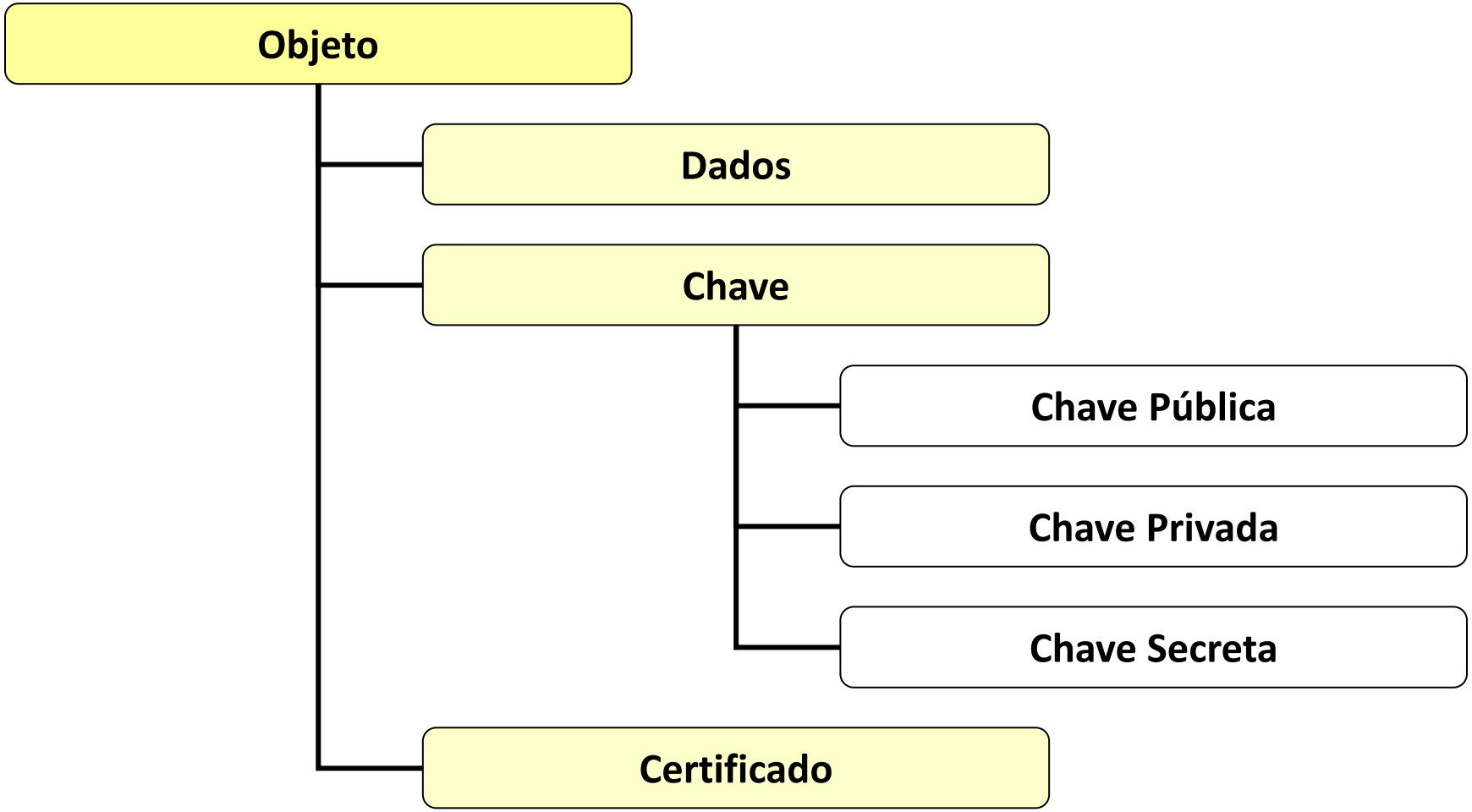
Serviços criptográficos do Smartcard: Middleware

- **Bibliotecas que servem de ponte entre as funcionalidades do Smartcard e as aplicações de mais alto nível**
- **Baseado em soluções normalizadas:**
  - PKCS #11
    - Cryptographic Token Interface Standard (cryptoki)
    - Definido pela RSA Security Inc.
  - PKCS #15
    - Cryptographic Token Information Format Standard
    - Definidopela RSA Security Inc.
  - CAPI CSP
    - CryptoAPI Cryptographic Service Provider
    - Definido pela Microsoft para sistemas Windows
  - PC/SC
    - Personal computer/Smart Card
    - Plataforma para acesso a smartcards em Windows e Linux

PKCS #11:
Integração do Middleware Cryptoki



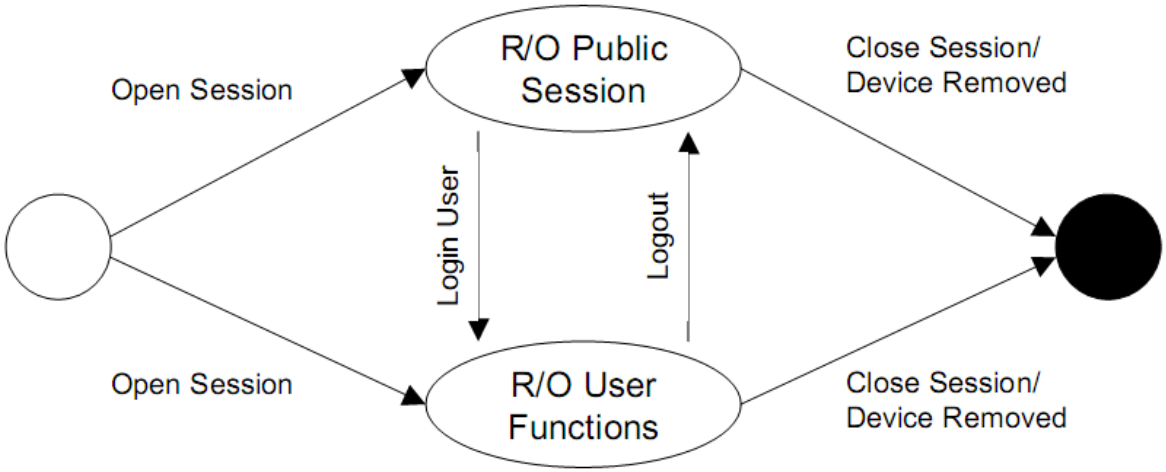
PKCS #11: Hierarquia de objetos



PKCS #11: Sessões do Cryptoki

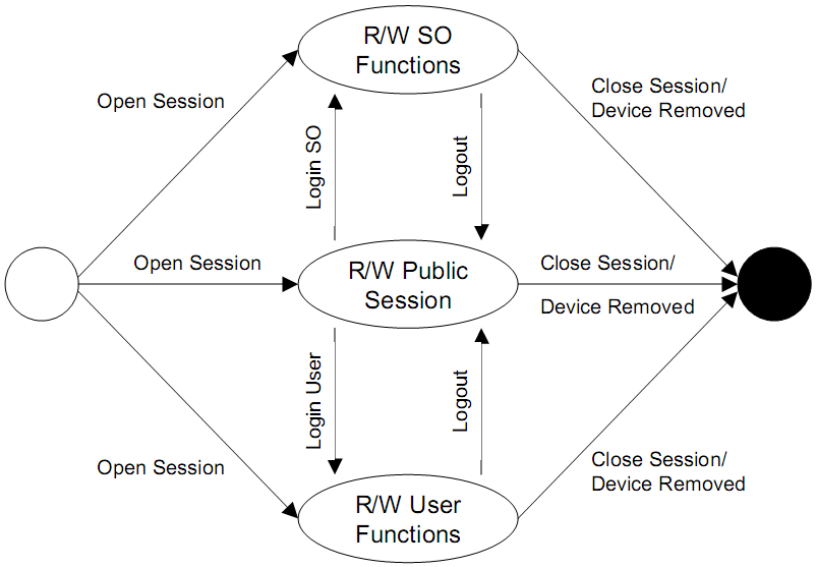
- **Ligações lógicas entre aplicações e cartões (tokens)**
  - Sessões de leitura
  - Sessões de leitura e escrita
- **Operações em sessões ativas**
  - Administrativas
    - Login/logout
  - Gestão de objetos
    - Criar ou destruir um objeto no cartão
  - Criptográficas
- **Objetos de sessão**
  - Objetos temporários criado (e válidos) durante a sessão
- **Tempo de vida das sessões**
  - Normalmente apenas para uma única operação

PKCS #11: Cryptoki Sessões de Leitura



- **Sessão pública de Leitura**
  - Acesso de leitura aos objetos públicos
  - Acesso de leitura/escrita aos objetos de sessão públicos
- **Funções de leitura do utilizador**
  - Acesso de leitura a todos os objetos do cartão (públicos ou privados)
  - Acesso de leitura/escrita a todos os objetos de sessão (públicos ou privados)

PKCS #11: Cryptoki Sessões de leitura e escrita

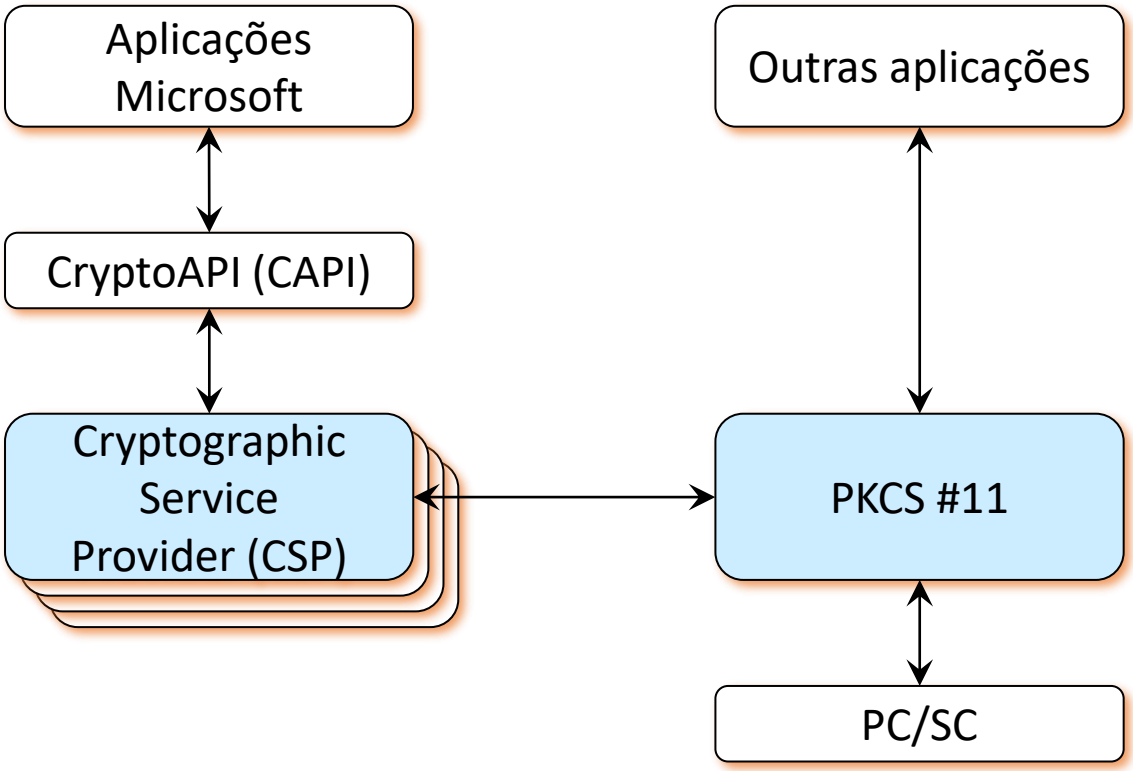


- **Sessão pública e Leitura e Escrita**
  - Ler e escrever todos os objetos públicos
- **Funções do SO de Leitura e Escrita**
  - Ler/escrever objetos públicos
    - Não os objetos privados
  - O SO pode definir o PIN dos utilizadores
  - SO = Security Officer
- **Funções do utilizador de Leitura e Escrita**
  - Ler e escrever todos os objetos

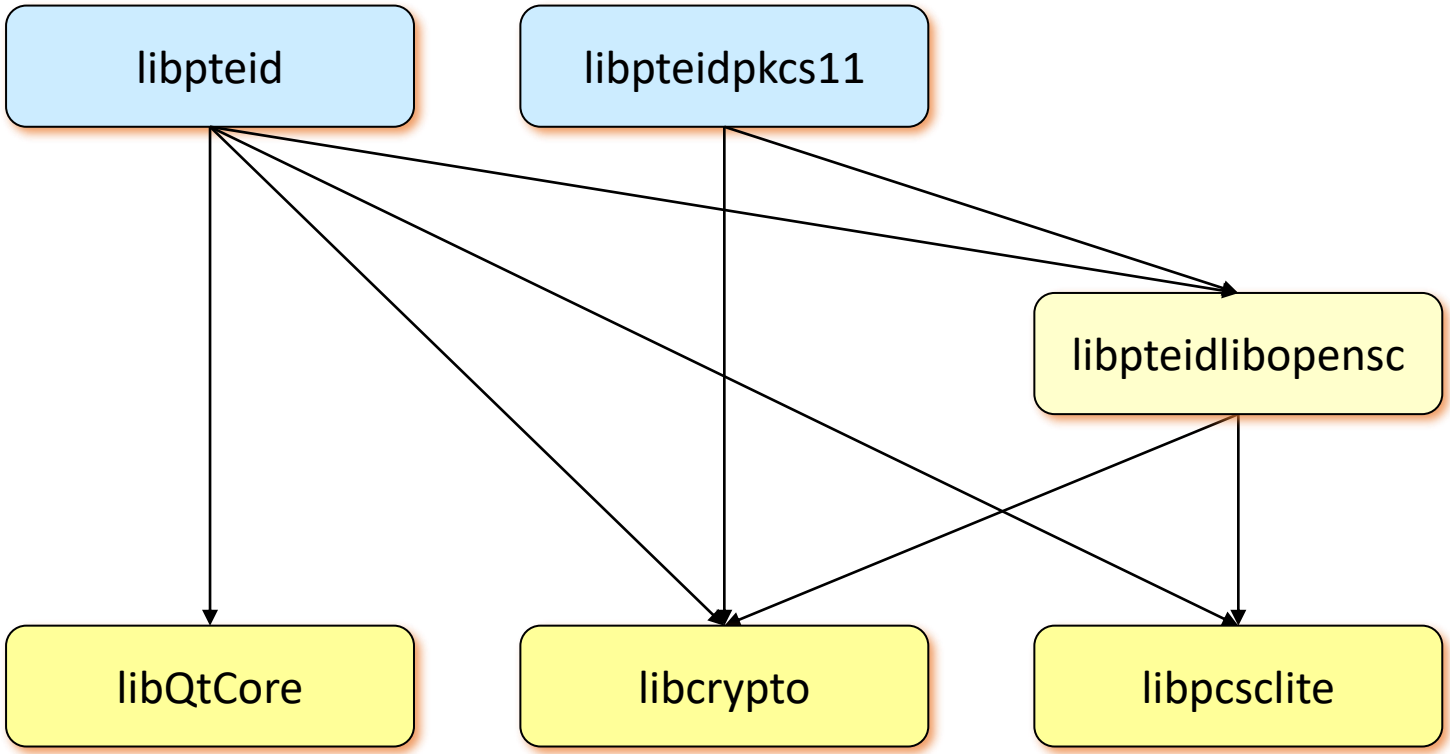
PKCS #11: Conceitos utilizados pelo CC

- **PIN de Autenticação**
  - PIN do utilizador no PKCS #11
- **PIN de Assinatura**
  - Não exposto pelo interface PKCS #11
- **PIN de Morada**
  - Não exposto pelo interface PKCS #11
  - 0000 por defeito nos cartões recentes
- **PKCS #11 SO PIN**
  - Não utilizado pelos titulares do cartão

Middleware PTEID para Windows



Middleware PTEID para Unix



PTEID middleware & SDK

- **Distribuição pública**
  - Windows
  - MAC OS X Yosemite
  - Linux
  - Caixa Mágica, Fedora, OpenSuse, Red Hat, Ubuntu
- **Linguagens**
  - Bibliotecas dinâmicas para C/C++
  - Wrapper Java (JNI) para as bibliotecas C/C++
  - Wrapper C# .NET para as bibliotecas C/C++
- **Manuais**
  - Validação de Número de Documento do Cartão de Cidadão
  - Autenticação com Cartão de Cidadão
  - Manual Técnico do Middleware do Cartão de Cidadão
  - Certificados e Entidades de Certificação
  - Outros

PTEID middleware & SDK

- **API adicional para interagir com o CC**
  - Fornecida pela biblioteca libpteid.so
- **Permite acesso ao dados relativos ao cidadão**
  - Nome, Fotografia, etc...
- **Objetos PTEID armazenados como ficheiros**
  - 3f000003 = Trace
  - 3f005f00ef02 = Citizen Data (Identification Data, Photo)
  - 3f005f00ef05 = Citizen Address Data (Pin Protected)
  - 3f005f00ef06 = SOd (Security Object Data)
  - 3f005f00ef07 = Citizen Notepad

Assinatura de Documentos

- **CC permite geração de assinaturas e estas podem ser inseridas em objetos**
  - Emails, **Documentos PDF**, ...

- **Assinatura digital substitui assinatura caligrafrada**
  - Importante no contexto legal ou Adm. pública (notas na UA)
  - Nativamente suportada em alguns formatos

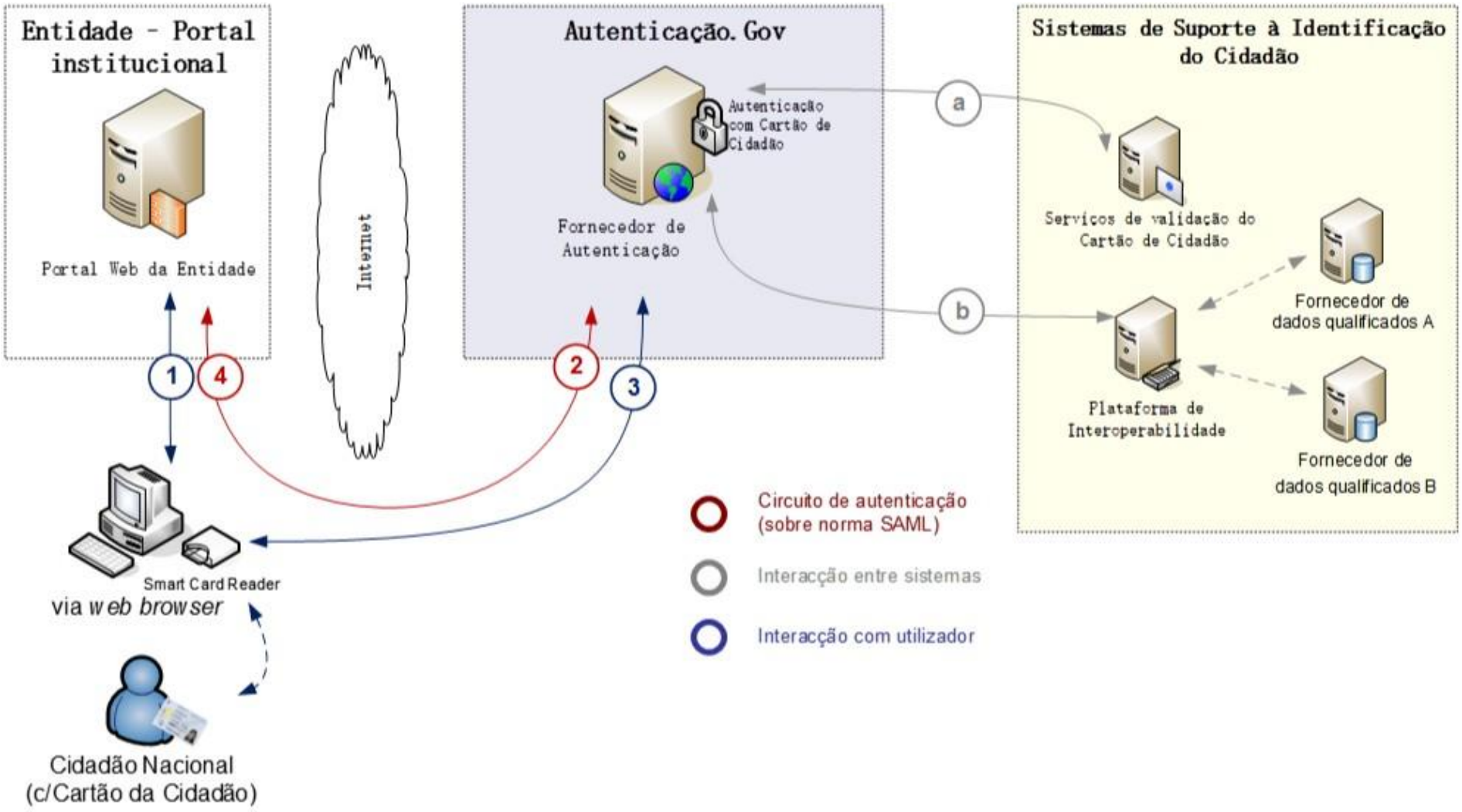
- **Utiliza chave privada e Selo Temporal da PKI**
  - CC: http://ts.cartaodecidadao.pt/tsa/server
  - Selo Temporal é vital para garantir instante da assinatura

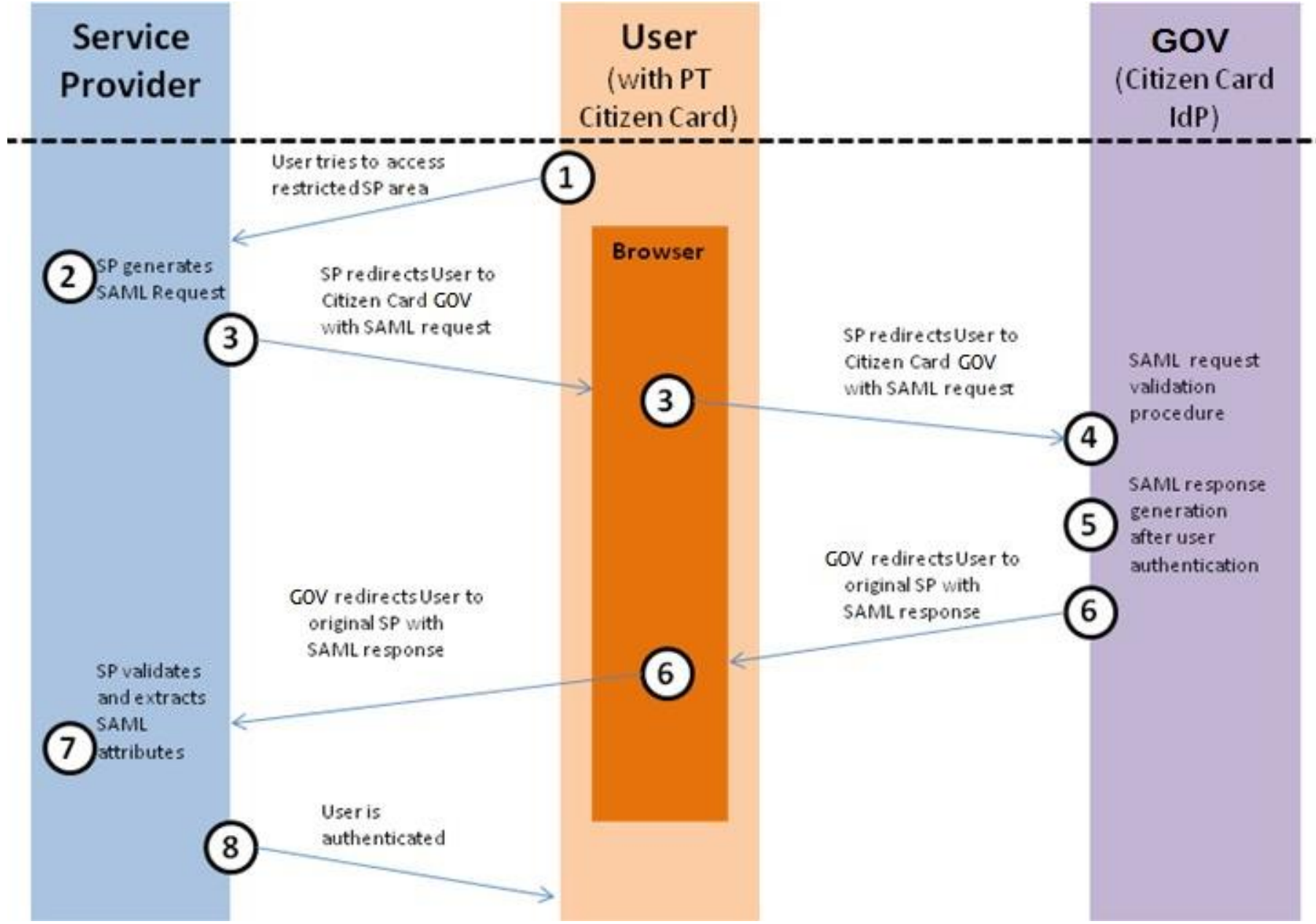
Autenticação com o CC

- **Autenticador envia um NONCE ao CC para ser cifrado com a chave privada**

- **Problema: Browsers não possuem acesso ao cartão**
  - Possível configurar libpteidpkcs11.so, mas só para acesso via API PKCS#11
  - Possível usar applet Java (obsoleto)
- **Solução: Utilizar um plugin no computador do utente**
  - Expõe servidor web no localhost
  - Permite acesso ao cartão através do servidor web
    - Apenas a pedidos autenticados pela infraestrutura do CC
  - Necessita de aprovação prévia para cada nova integração

Plugin Autenticação.gov




Service Provider
User (with PT Citizen Card)
GOV (Citizen Card IdP)
Browser
User tries to access restricted SP area
1
2
SP generates SAML Request
SP redirects User to Citizen Card GOV with SAML request
3
3
SP redirects User to Citizen Card GOV with SAML request
SAML request validation procedure
4
SAML response generation after user authentication
5
GOV redirects User to original SP with SAML response
6
GOV redirects User to original SP with SAML response
6
SP validates and extracts SAML attributes
7
User is authenticated
8

Chave Móvel Digital (CMD)

- **Objetivo: possibilitar autenticação/assinatura mesmo sem o CC presente**
  - mas com segurança de nível "semelhante"

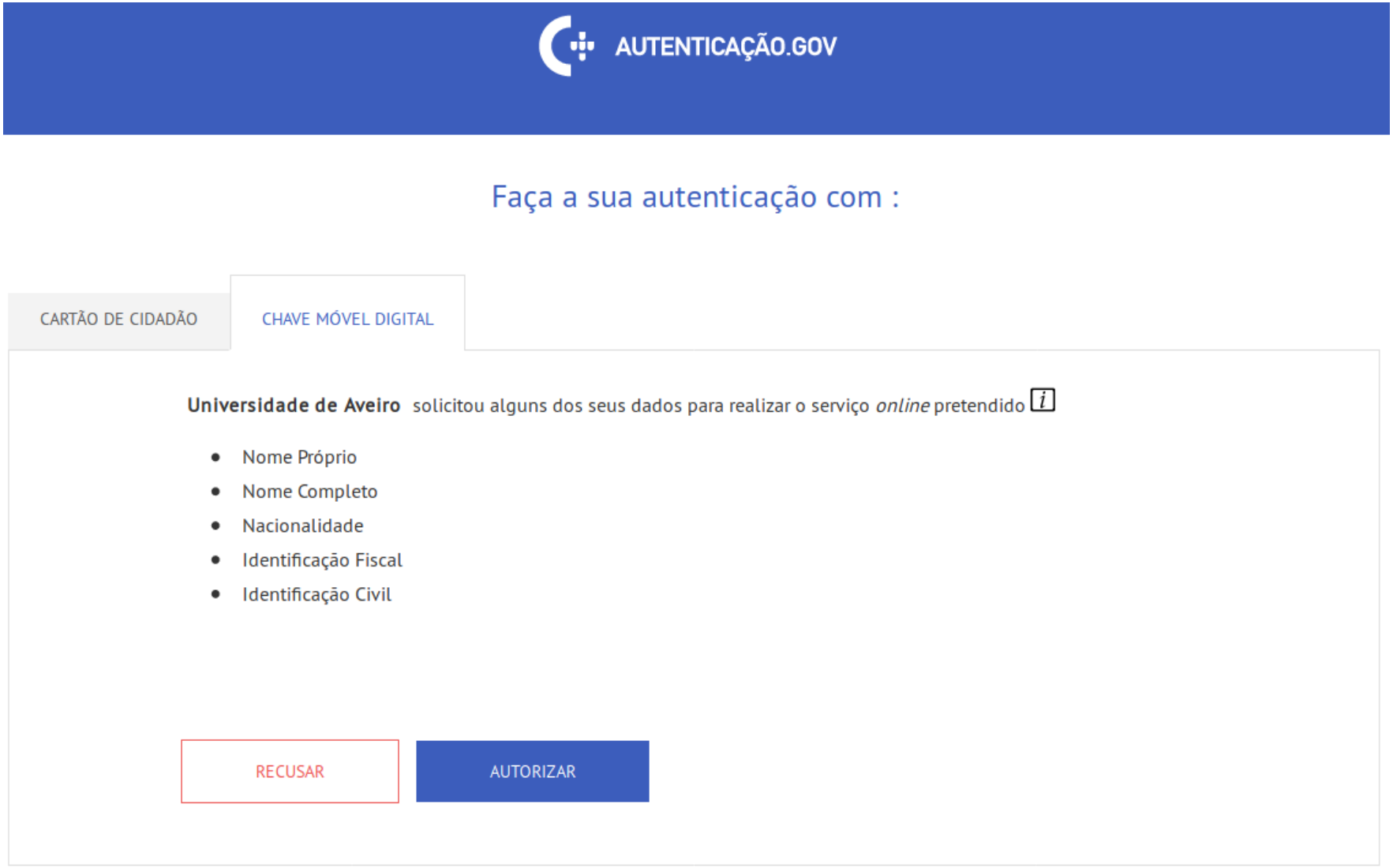
- **Princípios de funcionamento**
  - Necessita de um CC para autenticar o pedido de uma CMD
  - Utentes podem autenticar-se/assinar documentos usando a CMD
  - Não necessita de plugin instalado
  - Não necessita de cartão para utilização futura
  - Utiliza 2FA: PIN no site + código por outro canal (SMS, Twitter...)

Chave Móvel Digital

Processo baseado na criação de um par de chaves, armazenado remotamente

1. **Cidadão usa o CC para pedir uma CMD**
   1. Especifica uma senha/pin
   2. Especifica um canal de autenticação
2. **É gerado um par de chaves**
3. **Chave pública enviada para geração de certificados**
4. **Chaves e certificado armazenados em ambiente seguro**
   1. Protegido pela senha do utilizador
5. **Permitidas operações a quem validar a autenticidade**

Chave Móvel Digital



Chave Móvel Digital

Chave Móvel Digital